COMET. A Method for Analysing Collective Design Processes

Françoise Darses, Françoise Détienne, P. Falzon, Willemien Visser

Collection

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Method for Analysing Collective Design Processes

Françoise Darses - Françoise Détienne
Pierre Falzon - Willemien Visser

N° 4258
Septembre 2001

THÈME 3

Rapport de recherche

COMET
A Method for Analysing Collective Design Processes

Françoise Darses, Françoise Détienne, Pierre Falzon, Willemien Visser

Thème 3A
Projet Eiffel

Rapport de recherche n° 4258 – Septembre 2001

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)
Téléphone : +33 (1) 39 63 55 11 - Télécopie : +33 (1) 39 63 58 92

COMET: Une Méthode d'Analyse des Processus de Conception Collective

Résumé : Ce texte est centré sur la modélisation de la conception collective, en particulier sur les activités coopératives mises en œuvre dans des situations de co-conception. En psychologie cognitive, alors que des méthodes sont proposées pour l'analyse des protocoles individuelles, l'analyse des dialogues issues de situation collective est beaucoup moins bien outillée. Or, de nombreuses activités professionnelles sont mises en œuvre à travers des interactions verbales entre les différents acteurs. Nous proposons une méthode pour analyser les activités mises en oeuvre par des concepteurs au cours de réunions où ils travaillent ensemble sur des projets de conception (i.e. leurs dialogues et les productions et utilisations de représentations externes). Cette recherche a un triple objectif: modéliser les activités collectives (en particulier, le raisonnement), proposer une méthode d'analyse pour les psychologues cognitivistes et autres chercheurs intéressés par les activités collectives, étendre l'analyse des protocoles verbaux à des situations de travail collectives. Notre méthode sera illustrée à travers des données recueillies dans des études sur la coopération dans la co-conception dans deux types de tâches, la conception de logiciel et la conception de réseaux.

Mots-clés : méthode, activité de conception collective, analyse de dialogues, interactions verbales, coopération, conception de logiciel, inspection de logicial

COMET: A Method for Analysing Collective Design Processes

Abstract: This paper focuses on modelling collective design, especially co-operative activities in co-design situations. Cognitive psychologists have proposed various methods for the analysis of verbal individual protocols, but much less for dialogues in collective work settings. Many professional activities, however, are carried out by people working together through verbal interactions. A method is proposed for analysing activities of designers during the meetings in which they are working together on a design project (i.e. their dialogues, and their generation and use of external representations). This research has a triple objective: modelling collective-design activities (especially the reasoning component), proposing a data-analysis methodology for cognitive psychologists and other researchers interested by collective design, and extending verbal protocol analysis to collective work situations. The method will be illustrated through data collected in studies on the co-operation in co-design situations in two different collective-design tasks, software design and design of local area networks.

Keywords: Theoretical framework, method, collective design activities, dialogue analysis, verbal interactions, co-operation, software design, software review,

1 RATIONALE BEHIND THE COMET METHOD

The design of products is mainly achieved by teams, in which designers must cooperate to develop the solution. As any other professionals, designers need assistance tools, which can be either computational (CAD or CAM for instance) or methodological (project management tools, decision making methods, etc.). To be efficient, these tools must be able to support the cognitive processes that underlie the designers' activities. Thus, collective design cognitive processes must be identified and modeled. But it appears that more often than not, the cognitive dimensions of designers' work are directly extrapolated from observations of their activity or from interviews, without going through a phase of cognitive modelling. When transcriptions of verbal or visual material are made, it is generally as an anecdotal support for extrapolations of cognitive facts and phenomena. This approximative nature of the analyses makes cognitive system design more like the result of inspired innovation rather than the result of well-grounded methods. Cognitive Task Analysis (CTA) methods are thus needed to understand design activities and to develop tools for designers.

COMET is a method which provides a CTA technique for analysing and modelling collective design activities. The ensuing objective is to build cognitive models useful for intelligent assistance systems for designers. COMET is based on the analysis of the dialogues exchanged between designers in co-design situations. Cognitive psychologists have proposed various methods for the analysis of individual verbal protocols, but much less for dialogues in collective work settings. Many professional activities, however, are carried out by people working together in collective settings and through verbal interactions. A CTA method is thus needed for the analysis of these real-life task-oriented dialogues.

COMET gives some methodological guidelines related to a cognitive ergonomics approach of the activity. These guidelines have been elaborated and used in two case studies of co-design situations (computer network co-design and software review meetings). Some results of these studies will be given. COMET has also been applied and assessed in other design studies (Reuzeau, 2000; Cahour, 2001; Martin et al., 2001).

COMET takes place when the socio-technical macro-analysis of the activity has finished: the ergonomic problem to solve has been formulated, the concerned operators have been identified, the prescribed task has been understood through interviews, the organisational demands have been highlighted. COMET stems from observation of work in situ, and especially from observations of design meetings. The observation is passive: the analyst attends the design meetings, but is not supposed to intervene neither on a technical side (as a co-designer, expert in ergonomics) nor on a mediating side (with the role of connecting all the parnter's views). The verbal protocols collected through this way provide the ergonomists with inputs for applying COMET, and then analyzing and modelling the collective design activity. The method is based on a logico-temporal analysis of the data. The idea here is to preserve the trace of the rationale that guides operatives' actions in order to analyse the types of reasoning process at work. To do so, one essentially relies on detailed analysis of audio and video recordings of the activity, from which one transcribes at length the verbal communications. COMET starts from this point It provides a coding scheme, based on a predicate-arguments pattern, from which the cooperative processes between designers can be formalized.

2 THE ANALYSIS OF VERBAL DATA

COMET is based on the analysis of dialogues collected by audio or video recording and transcripted into verbal protocols. In the following sections, we show that, up to now, protocol analysis methods have mostly been used to analyse individual activities. The existing methods of dialogue analysis are more on the linguistic side. Dialogues are formalised at the micro-level of linguistic discursive processes, which is irrelevant for the cognitive ergonomics purposes. The situated and task-oriented context of design activities – as any other professional activity – makes necessary to take into account task components.

2.1 Individual verbalizations

Protocol analysis is based on the assumption that much information about cognitive processes can be accessed by asking subjects to "verbalize their thoughts" or to "think aloud" (Ericsson & Simon, 1984). Subjects are asked to report all pieces of information they take into consideration: the choices they face, the criteria they use for decision, their reasoning, their hesitations, their re-consideration of past decisions, etc. This method is not *introspection*: one does not want the subjects to speculate and theorize about mental processes, nor to give comments or opinions about their activity. It is not either *explanation*: the subjects are not requested to transmit their knowledge to the analyst. If explanations are required in order to understand a subject's verbalizations, they should be requested later. Instructions provided to the subjects should focus on obtaining the stream of consciousness of the subject while completing the task.

The validity of individual verbalisations mainly depends on four factors:

- *the naturalness of a solitary problem solving process in a limited amount of time*: some problems require the involvement of a team, and may last for much longer than the time generally available for data collection;

- *the level of difficulty of the task*: highly demanding tasks, complex problems require an intense cognitive activity. Concurrent verbalization, which constitutes a secondary task, may then hinder performance;

- *the level of constraint on verbalization*: the presence of an experimenter can lead the subjects to justify their assertions or decisions. This added constraint may provoke a reflexive process which may bias the reasoning and the subsequent verbalization;

- *the degree of automatization of the activity*: routine tasks are fulfilled more or less automatically. A request for verbalization makes it necessary to decompile mental rules, if possible. The validity of the data obtained in such cases should thus be questioned.

Individual verbalizations are not directed towards someone[1] (although the subject might consider the experimenter – in spite of the instructions - as an interlocutor). We will consider that these verbalisations do reveal, in its weak sense, the *argumentative* process, insofar as one defines this term as the global *reasoning* developed to go from a goal to a solution. Thus, these verbalizations are supposed to provide information about the current task, the focus of the next task, the current state of solution, the strategies, the domain rules, etc. Cooperative dialogues differ from these verbalisations, both in their nature and in the CTA method that can be used. This is highlighted in the next section.

2.2 Cooperative dialogues

Some authors, as Goldsmidt (1995), say that « thinking aloud and conversing with others can be seen as similar reflections of cognitive processes », and accept them as « equal windows into the cognitive processes involved in design thinking » (p.193). We do not follow this opinion, and we claim that verbal data obtained through cooperative dialogues differ from those obtained in individual verbalisations regarding: (i) the ecological validity, (ii) the reactivity, (iii) the argumentation level, (iv) the management of the common frame of reference, (v) the level of explicitation, (vi) some psychosocial factors.

- *ecological validity:* Cooperative dialogues are interactions who occur in tasks that require collective problem solving. Subjects have to interact in order to fulfil their job, and the presence of the analyst in such collective situations is easily forgotten. The biases mentioned in § 2.1 are

[1] Of course, some individual verbalizations, such as speeches, are directed towards someone and have an argumentative nature which is developed for convincing the interlocutor. But these discourses are not those focused on when analysing task-oriented individual verbalizations.

avoided. Additionally, the analysis of such dialogues gives access to the mechanisms of cognitive cooperation.

- *reactivity:* Reactivity is defined by Brinkman (1994) as the modification of cognitive processes due to the concurrent or retrospective verbalisation. In an individual situation, verbalisation is an additional task for the designer and thus may lead to some more or less important transformations/deformations of the activity with respect to the one that would have been implemented in a situation without such an added task of verbalisation. In the collective situation, cooperative dialogues are “natural”, in that they do not constitute an additional task. To the contrary, it is a very important part of the activity. Cooperative dialogues reflect the contribution of each designer to the collective design activity.

- *argumentation level:* The reasoning process in cooperative dialogues is shared, questioned and built by the team. Here, the argumentative process must be understood in its strong sense, that is to say, as the expression of arguments aimed at acting on the interlocutor’s representations and actions. The dialogues dealt with are not free from factors outside the "purely" cognitive dimension. People want to put forward their opinions not only from a task-related standpoint, but also because they have various personal or organizational reasons to do so. Because of these socio-cognitive factors, the speaker may not verbalise his/her real thoughts, according to sociological strategies (face management, professional or personal identity stakes, conflict involvement, etc.).

- *management of the common frame of reference:* the shared knowledge, or common frame of reference (Hoc, 2000; Terssac & Chabaud, 1990) is composed of general domain knowledge (concepts, theories, processes, procedures, know-how) and of circumstantial knowledge (state of a given problem solving process). It may vary according to the expertise of the partners (expert-novice differences), the circumstantial knowledge (project history), the degree of multidisciplinarity (multi-expert decision making) and the degree of reciprocal personal knowledge. In cooperative situations, task completion depends on successful (that is both efficient and productive) interactions between the partners. A way to improve efficiency and productivity is to enlarge the common frame of reference. Thus, part of the interaction process is devoted to the management of the common frame of reference. It has been noted for instance that the information provided in the course of problem solving sometimes extends well beyond what is needed to satisfy immediate goals: subjects take advantage of the opportunities of the issue being discussed to "teach" their partners. Thus, the management of the common frame of reference is additional to the fulfilment of the task.

- *level of explicitation:* Subjects interacting in professional problem solving situations share a certain amount of knowledge. What they say is what they judge useful and sufficient to say considering their partners' knowledge. As stated by relevance theory (Sperber & Wilson, 1989), subjects' utterances are determined by the inferences they wish to induce in their partners' cognitive system, and thus depend on the model they have built of their partners' knowledge. Thus, a large part of the underlying reasoning will remain implicit, which the dialogues will not reveal. This is why dialogue analysis often necessitates the help of the participants in order to decipher the exact motivations or to validate the interpretations of some utterances. Subsequent interviews are then necessary.

- *psychosocial factors*: Team design work is a social activity, in which psychosocial factors may influence the cognitive processes and the cooperative dialogues. This was shown for instance in a case study presented by Sauvagnac & Falzon (1996). In order to study cooperative diagnosis, the authors collected the dialogues between maintenance and production operators in agrofood industry. But some reflexive assessments about these dialogues - supposedly cooperative - highlighted that the organizational and social problems between services were hidden under the dialogues, and influenced much the task performances. Individual verbalisation does not suffer from this possible influence.

2.3 The analysis of task-oriented dialogues

The analysis of dialogues has been for long investigated by linguistics, especially in pragmatics. In task-oriented activities of design, the dialogues are said to be cooperative, since the partners share a common goal: they have to converge towards an agrement about a solution. This differs from some other types of dialogues, such as political debates, interviews, chattering, etc., where the aim is not primarly to collaborate towards a common production.

To formalise these task-oriented dialogues, we have adopted an interlocutory analysis (Brassac, 1992; Ghiglione & Trognon, 1993), which is based on the speech acts properties. This notion, extended from the speech acts theory (Austin, 1962; Searle, 1969), refers to the socio-cognitive dimension of the dialogues. It stems from the idea that speaking is far from being a simple information transmission: it is also a way of acting on the world and acting with others, it is a way of defining structures for exchanges and transactions between partners (Brassac & Gregori, 2001). The illocutory words' property defines a type of *acting* (for instance, *request* or *inform*), which can be interpreted and formalized when analysing protocols. For instance, if a designer says: « This element is too expensive. I suggest this other one », he/she is obviously *acting* on the design process by *critiquing* a previous solution and *proposing* an alternative one, on the basis of a price criterium.

COMET is based on such an interlocutory analysis. It uses a predicate/argument format, where the predicates correspond to types of speech acts (see section), which is applied to the design objects (the solution, the domain rules, etc.). This formalisation has been used in cognitive ergonomics but mostly as tailor-made methods for a specific situation within a specific study (Dorst, 1995; Cross et al., 1996). Our purpose is to provide searchers and practitioners (ergonomists, psychologist, engineering designers, etc.) with a CTA method which will not have to be each time reinvented, but only customised according to the case specificities. COMET was built in previous studies (Falzon & Darses, 1992; Darses et al., 1993; Darses & Falzon, 1994; D'Astous et al., 1998). A similar CTA method, aimed at modelling the cognitive activities in the domain of dynamic situations, was developed by Amalberti & Hoc (1998.) and Hoc & Amalberti (1999). The authors propose to formalise three keys mental activities (information elaboration, decision-making and diagnosis) through a coding scheme using the predicate-arguments formalism, where predicates code elementary cognitive activities and arguments express mental representations of activities. The method COMET presented in this paper is tailored for the collective design activities.

3 CO-OPERATIVE PROCESSES IN CO-DESIGN SITUATIONS

Collective practices of design have been studied for a long time from multiple viewpoints: organisational analysis, social psycho-social and psychic analysis (focused on the role of factors such as the degree of trust in the others, the recognition of personal competence, personal development through work, power redistribution, necessity of protection, etc.). The approach chosen by cognitive ergonomics differs from these types of analysis. While acknowledging that the dimensions described above obviously affect collective work and the development of a design activity, we have chosen to focus more on the cognitive aspects of collective design, and especially the cognitive aspects of the co-operative processes occurring during co-design.

The actors involved in a design process are not all involved in the same way - some are involved in ***co-design*** activities while others participate in ***distributed design*** activities (Darses and Falzon, 1996; Darses et al., 1996; Visser, 1993). These two situations can be found during one single design process and can also be taken in charge one after the other by one single actor. This distinction between co-design and distributed design is useful since each of these forms of collaboration during product development induces different collective processes.

3.1 Design Problem solving

Cognitive ergonomics does not identify design in relation to a social function or a status, but qualifies as ***design*** tasks certain professional activities in which a set of formal characteristics can be identified. Therefore, one can identify numerous professional domains that deal with design. It can be the design of material artefacts (e.g. mechanical engineering, electronics, and architecture) or the generation of symbolic or abstract devices (e.g. planning, computer programming or resource allocation). The features of design tasks are now well known (Goel and Pirolli, 1989; Simon, 1981):

- Problems are large and complex. They are generally not confined to local problems, and the variables and their interrelations are too numerous to be divided in independent sub-systems.
- One consequence of this complexity is that the resolution of these problems often requires that multiple competencies be put together, which leads to development of collaboration within a single work group.
- There are many degrees of freedom in the problem initial state. This has led to consider design problems as ill-defined problems.
- A design problem has several acceptable solutions, not only one.
- There is not any pre-determined way that leads to the solution. A certain number of useful procedures and design methodologies are known, one can refer to similar projects already studied or to existing prototypes, but, each time, it is necessary to reinvent the steps that separate specifications and production.
- The definition of the problem and the elaboration for the solution are made in interaction. The problem does not exist before the solution - both are built simultaneously.
- The evaluation of the solutions is delayed as much as possible or in any case limited to the final solution. This is because the generation of all alternative design solutions is costly or impossible to carry out and also because objective metrics do not exist. Therefore, final solutions are satisfactory but not optimal.

One important objective of cognitive ergonomics has been to understand how these tasks characteristics determine the building of the problem-space explored by the designers. From this basis, design tasks have been in a first time studied in their individual practices. But the growing importance of collective design situations makes necessary to understand the designers' activity from a collective standpoint. The collective design situations are of two complementary types: *distributed design* situations and co-design situations.

In ***distributed design*** situations (Béguin, 1994; Karsenty, 1994), the actors of the design are simultaneously, but not together, involved on the same collective process. They carry out well-determined tasks, which have been allocated beforehand to them, and they pursue goals (or at least sub-goals) that are specific to them. They have as an objective to participate as efficiently as possible in the collective resolution of the problem. Distributed design is typical for concurrent engineering, in which the various sides of the production system must function in strong synergy during the product-development cycle.

In the ***co-design*** situations, design partners develop the solution together: they share an identical goal and contribute to reach it through their specific competencies; they do this with very strong constraints of direct co-operation in order to guarantee the success of the problem resolution. The competence of the partners can vary depending on the level of competence (e.g. interaction between designers of different seniority) or on the type of competence (e.g. interaction between drafters and engineers). Co-design has been studied by different authors (D'Astous et al., 1998; Karsenty, 1994; Malhotra et al., 1980; Visser, 1993). The two studies presented in this paper deal with co-design tasks.

A crucial mechanism in co-design is *cognitive synchronisation.* Cognitive synchronisation enables the partners to reach two objectives: (i) to make sure that each has knowledge of the facts relating to the state of the situation (i.e. problem data, solution states, accepted hypotheses, etc.); (ii) to make sure that they share a common knowledge regarding the domain (i.e. technical rules, objects in the domain and their features, resolution procedures, etc.). Cognitive synchronisation therefore aims at establishing a context of mutual knowledge, at building a common operative system of reference (Karsenty & Falzon, 1992; Terssac & Chabaud, 1990). Cognitive synchronisation activities will vary depending on the amount of shared knowledge. This means in particular that the parity or non-parity of the dialogue (dialogues between pairs vs. expert/novice dialogues or dialogues between subjects with distinct knowledge) will have an effect on the necessity to communicate general knowledge. It has been shown before (Falzon, 1991) how the common-knowledge hypothesis in this domain in dialogues between experienced partners enables an economy in communication by using operative languages, and how partners used repair dialogues when this hypothesis is at fault, these repair dialogues aiming precisely at levelling general knowledge. The necessity to assure the nature of the common operative system of reference leads each partner in a dialogue to build a model of the other, as has been shown in various studies (Cahour & Falzon, 1991).

4 THE COMET METHODOLOGY

In this section, we present COMET. This method provides some guidelines for the analysis of real-life task-oriented dialogues. These guidelines provide a frame for analysing design dialogues in real situations. The frame is made up of two distinct and hierarchical levels of coding: ***basic level*** (where individual UNITS are identified) and ***composite level*** (where the CO-OPERATION MOVES are identified).

4.1 The basic level: coding individual UNITS

4.1.1 First phase: identifying turns in the protocol

A dialogue is transcribed into a verbal protocol. Each protocol is cut up into a series of individual participant utterances ("turns"), according to change of locutor. The gestural and physical-movement aspects of the interaction have not been addressed in these studies. What is analyzed is the verbal material.

Designer	Verbal data
B	Why did you put 150 there ?
M	I don't believe in using 150 DEFINE.
M	These will do the same thing, but the compiler will check them while the compiler doesn't check DEFINEs.
C	There may be more than 50 error messages, you know !
M	Ah no, this is just a type, like the type of the message itself.
C	Ah ah
M	It's just that I need, I need some fields OK, these four fields there !
M	Because I need some fixed rrays at the start for the messages.
M	So, I fix them, I fix the first four.
M	The additional messages will follow.
M	We'll be able to put what ever we want, an error message, insufficient memory ...
B	Why, then , if we can use them anyway !
B	OK, we don't have the choice !

Table 1 Excerpt of a design dialogue

4.1.2 Second phase: coding turns into individual UNITS

These utterance turns are cut up further into one or more individual UNITS according to a coding scheme developed on a Predicate(Argument) basis. Each predicate admits a number of possible arguments, but not any argument. Predicates (ACT) correspond to actions implemented by participants; arguments (OBJ) correspond to objects related to the action. According to the form of the predicate (Assertion or Request), each unit is modulated (MOD). The default value of a unit is assertive: modulation is coded explicitly only if its predicate is a request.

Thus, each UNIT is coded as MOD[ACT/OBJ], where MOD may be absent —in which case it is assertive (see Table 2).

Modulation (MOD)	**Predicate (ACT)**	**Argument (OBJ)**
Assertion Assertions are not coded as such: implicitly, a unit is assertive.	**Generate** (GEN) Proposing a new element into the dialogue (a solution, a goal, an inferred data, etc.)	**Problem data** (DAT)
Request (REQ)	**Evaluate** (EVAL) Judging the value of a subject. This evaluation can either be negative, positive or neutral.	**Solution elements** (SOL)
	Inform (INFO) Handing out new knowledge with respect to the nature of a subject	**Domain objects** (OBJ)
	Interprete (INT) Expressing a personal representation of a subject. This representation is made through the use of expressions such as "I believe that...", "I think ..." or "...maybe...".	**Goal** (GOAL)
		Domain rule or procedure (PROC)
		Task (TASK)

Table 2 Basic coding scheme, presenting the elements of each category

Utterances *	**Coding into UNITS**
A MPR is a connexion box	[INFO/ OBJ]
When is the minimum spanning tree to be applied ?	REQ [INFO/ PROC]
The building has two floors	[INFO/ DAT]
I propose 2 thin segments	[GEN/ SOL]
You need some filtering	[GEN/ GOAL]
Which aspect of the problem should be tackled now ?	REQ [GEN/ GOAL]
You forgot to take the loops into account when you computed total length	[EVAL - / PROC]
The MPR has 6 I/O, am I right ?	[EVAL/ OBJ]

Table 3 Examples of UNITS
* (this is not a dialogue, but isolated utterances)

4.2 The composite level: coding CO-OPERATION MOVES

4.2.1 First phase: grouping UNITS into sequences

At the ***composite level***, frequent and consistent UNITS are grouped into sequences. Such sequences can be formed in various ways, on a qualitative or on a quantitative basis, i.e. on a concept- or a data-driven basis. At this stage, these configurations are not named: this will be done during a second phase, when coding them as CO-OPERATION MOVES. It is to be noted that these sequences may cover utterances of two partners. This is not always the case. Sequences may also consist of a succession of UNITS emitted by a single partner.

Psychological expertise may guide a qualitative approach in a concept-driven way. Cognitive models of design (Visser, 1991, 1992) suppose, e.g., that evaluation proceeds by criticism of a solution proposal, possibly preceded by a request for such a criticism and backed up by a justification. The criticism has to be accepted or rejected, implicitly or explicitly. According to this example, a cognitive model leads us to search in the design protocol for evaluation sequences composed of these different activities.

A data-driven quantitative approach may be used to identify significantly frequent configurations of UNITS. Different types of method may be suitable for this aim. A statistical method grounded in information theory is Lag Sequential Analysis (LSA) (Allison & Liker, 1982). LSA enables the identification of UNITS that follow each other, with or without other UNITS in-between. The analysis consists in determining whether or not the frequency of a given unit is independent of the frequency of another one. Sequential structures enable the definition of configurations. Hierarchical clustering (Johnson, 1967) is another statistical method, which allows similarities inside sequential structures to be quantified.

On the one hand, statistical methods can be used to validate groupings that were identified using "pure" qualitative analysis (as presented above); on the other hand, cognitive models may be of help in interpreting and conceptually validating the configurations identified by statistical approaches.

A combination of qualitative and quantitative approaches is to formulate, on the basis of psychologically inspired hypotheses, grammatical rewriting rules (Gonzalez & Thomason, 1978) of sequences of UNITS, and then apply a statistical method on the result. Such a combined approach can be applied in an iterative way, in several cycles, as long as statistically significant configurations are detected.

The table below shows some sequences which were identified in design dialogues.

A sequence identification: Negative criticism of a solution, following a solution generation (and sometimes an explicit request for criticism), followed by a generation of an alternative solution	
Designer 1: I propose 2 thin cable. What do you think of this ?	REQ [EVAL/ SOL1]
Designer 2: This is not possible, because this type of cable is not long enough to fit here.	[EVAL - / SOL1]
Designer 2: You'd better use a thick one.	[GEN / SOL2]

Table 4 Identifying a sequence as a consistent episode of UNITS

4.2.2 *Second phase: formalising the sesquences as CO-OPERATION MOVES*

The sequences, identified as described in the previous section, correspond to basic co-operative interactions. We call these interactions CO-OPERATION MOVE. A CO-OPERATION MOVE names a sequence. For instance, the sequence presented in table 4 will be named {NEGATIVE EVALUATION}. The sequence of UNITS presented in the following table shows a {COGNITIVE SYNCHRONISATION} CO-OPERATION MOVE.

A sequence corresponding to the CO-OPERATION MOVE: { PROVIDING ADDITIONAL INFORMATION ON THE SOLUTION }	
Expert 1: So, I fix them, I fix the first four.	[INFO / SOL1]
Expert 1: The additional messages will follow	[INFO / SOL1]
Expert 1: We'll be able to put what ever we want, an error message, insufficient memory	[INFO / SOL1]
Expert 2: Why, then , if we can use them any way !	REQ [INFO / SOL1]
Expert 2: OK, we don't have the choice !.	[EVAL + / SOL1]

Table 5 Example of a CO-OPERATION MOVE IDENTIFICATION

The table below presents the list of possible CO-OPERATION MOVES provide by COMET.

- Enriching a solution
  Spontaneous solution generation by the expert (spontaneous means: not preceded by an explicit request from the partner), following information on problem data, or a solution generation, or information on a domain object
- Executing a task
  Solution generation following an explicit request from the operator
- Assisting problem representation
  Information generation about the problem following information on problem data
- Orienting problem processing
  Generation of a focus of resolution, following a solution generation by the less experienced operator
- Guiding problem description
  Request by the expert for specific problem data, followed by the provision of this data by the partner (this BCI occurs mostly at the beginning of the dialogue)
- Assisting goal planning
  Generation by the expert of problem processing goals, followed in some cases by preventive information (rules to be applied, facts abouts domain objects)
- Assisting meta-planning
  Request by the expert for a solution generation (following the description of the problem, or the criticism of a solution)
- Positive evaluations
  Positive criticism of solution, following a solution generation (and sometimes an explicit request for criticism), sometimes followed by preventive information or a generation of solution (solution expansion)
- Negative evaluations
  Negative criticism of a solution, following a solution generation (and sometimes an explicit request for criticism), sometimes followed by preventive information or a generation of solution or of goal
- Mitigated evaluations
  Globally positive criticism of a solution, following a solution generation (and sometimes an explicit request for criticism), followed by a modification of the solution ("yes, but" situations)
- Providing additional knowledge or correcting knowledge
  Information on domain rules or object, following an explicit request or an inexact assertion on a rule or an object
- Providing additional information on solution
  Information on solution elements under development. This allow paticipants to make sure they share a common representation of the current solution.

Table 6 CO-OPERATION MOVES provided by COMET

5 APPLYING COMET: WHICH ISSUES, WHICH LIMITS ?

5.1 Issues concerning the UNITS coding scheme

5.1.1 Defining the coding rules: an iterative process and a reciprocal assessment

The first question is to define predicates and arguments. In the two studies, this definition was accomplished through an iterative process: the analysts coded separately a same dialogue extract and then confronted their coding. It is to be noted that two questions are discussed at this point. First, the level of granularity of the coding scheme (what is the size of the smallest unit?); second, the nature of predicates and arguments. The analysts decide on some rules, choose a second dialogue extract and then iterate the process.

One point needs to be stressed here: it is important that the analysts keep a precise trace of their own dialogue and decision processes. For instance: Why did they finally decide not to use this predicate but rather that one? What is exactly encompassed by this particular predicate? Ideally, the coding rules should be written down after these discussions. By "coding rules", we do not only mean the coding scheme, but also the rationale behind the scheme. As a matter of fact, we noted that analysts might very well converge on a coding scheme without being totally conscious of the rules they actually use. This phenomenon becomes obvious when asking a third person to code a dialogue[2]. Writing down the rules has two benefits. First it helps the analysts in formalising the scheme; second it facilitates the transfer of the coding scheme towards a third analyst if needed.

5.1.2 Dealing with the argumentation level: acknowledging the argumentative role of basic UNITS but coding this role further in the analysis

An important issue at this stage is also to determine a position relative to argumentation. Consider the following extract, taken from a "network design" dialogue. A designer has proposed to use the ceiling to install the cables. The other designer then says:

Designer 1:	*I think we should stick to using the existing shafts.*	1 [GENERATE (SOLUTION)]
Designer 2:	*The ceiling can be used*	2 [GENERATE (SOLUTION ALT)]
	but this brings about additional labour and high installation cost"	3 [INFORM (PROCED. KNOWL.)]

Table 7 Coding the knowledge level

The second sentence is a proposal for an alternative solution. But it also has obviously a role of negative evaluation of the first solution : additional labour and expenses justify the proposal to use existing shafts. Thus one coding solution would be to add the implicit negative value to the (2) UNIT, and transform it into an [EVALUATE NEGAT.] predicate and to create a [JUSTIFY] predicate, which status is attached thanks to the previous UNIT (3).This latter possibility is described in table 8 below.

Designer 1:	*I think we should stick to using the existing shafts.*	1 [GENERATE (SOLUTION)]
Designer 2:	*The ceiling can be used*	2 **[EVALUATE NEG** (SOLUTION)]
	but this brings about additional labour and high installation cost"	3 **[JUSTIFY** (PROCED. KNOWL.)]

Table 8 Coding the argumentative level

However, in this case, coding is not based on the intrinsic signification of the coded unit, but on its relation to a preceding one. As a matter of fact, it depends of the interpretation of the general context: the second sentence may play a justification role in a particular context (see table 8), but it may be a simple transfer of knowledge in another context (see table 7).

The alternatives are then either to code argumentative roles directly, i.e. *at the basic level* or to code argumentation only *at the composite level* of analysis. Different choices have been made in the two studies. We believe that the second alternative should be preferred. In the example presented above, at the basic level, an [INFORM] predicate is used; at the composite level, an argumentative role is taken into account and the UNIT corresponding to the [INFORM (PROCED. KNOWL.)] corresponds to the -OPERATION MOVES {JUSTIFICATION}.

5.1.3 Solving the knowledge ambiguities with the help of a domain expert

As was mentioned, the dialogues are technical dialogues between professionals, who share much domain knowledge. This has methodological consequences. Coding cannot be achieved without help from a competent professional; interpretation requires domain knowledge. Consider for instance the following utterance (taken from the "network design" dialogues):

"These Suns have Ethernet cards"

This can be interpreted either as:

"all Sun computers are equipped with Ethernet cards"

or as

"these particular Suns are equipped with Ethernet cards".

[2] As a matter of fact, asking the intervention of a third person can be used as a heuristic to check that rules have been explicitly stated.

In the first case, it will be interpreted as transfer of general knowledge; in the second case, it will be interpreted as transfer of information about the particular situation. The assistance of a domain expert is thus requested to solve this ambiguity. In the first case study related in this paper, (the"network design" study, see §6), coding of the protocols was checked with a domain expert. In the second case study (the "software review" study, see §6), the analysts elaborated the coding scheme, then explained it to a domain expert who did the actual coding of the dialogues.

5.2 Issues concerning the CO-OPERATION MOVES coding scheme

5.2.1 Argumentation

In the example presented in 5.1.3, "Justify" is not an element of the initial coding scheme. It may stand for several kinds of informative utterances (concerning procedures or domain-object characteristics, for instance). These utterances, in that specific context of {NEGATIVE-EVALUATION} move, play the argumentative role of "Justification". Thus, the creation of CO-OPERATION MOVES implies the identification of UNITS that can play a similar argumentative role in a specific position within a move. In the example above, the "Justification" role can be played, for instance, by an (INFORM /GOAL) UNIT. Another example is the "Prevention" role that appears in the {POSITIVE EVALUATION} CO-OPERATION MOVE, which consists in providing, after an initial positive evaluation of a partner's proposal, an information that may help the application of the positively evaluated solution ("*OK, but pay attention to...*"). Again, this information can take several forms of UNITS, such as (INFORM /PROCEDURE), (INFORM /DOMAIN OBJECT PROPERTY), or (INFORM /PROBLEM DATA).

5.2.2 Inferring missing UNITS

Design dialogues are task-oriented. The participants in such dialogues know what they may expect from each other and what others expect from them. This common frame of reference allows them to interpret what is said and to behave in an appropriate way. For instance, if A proposes a solution, A expects the solution to be evaluated by the others, even if A does not explicitly request it. Similarly, if A proposes a solution, and if B then proposes an alternative solution, A may interpret B's proposal as conveying an implicit criticism (a negative evaluation of A's solution). Or, if A proposes a solution, and if B then proposes an expansion of this solution, A may infer that B implicitly supports A's proposal. Design dialogues thus follow principles of operational co-operation. It is important to understand these principles in order to identify correctly the CO-OPERATION MOVES, either to define as optional some UNITS (for instance, evaluation requests), or to infer some missing UNITS (for instance positive or negative evaluations).

5.2.3 Sequence reorganisation

Verbal expression has to be linear, whereas relations between the underlying elements are not necessarily. The expression of the arguments "justifying a (negative) evaluation", e.g., may precede or succeed the expression of the evaluation. JUST-EVAL and EVAL-JUST are two possible surface forms of the same underlying argumentative movement OPINION-ARGUMENTS.

6 USING COMET FOR ANALYSING COLLECTIVE DESIGN: RESULTS FROM TWO CASE STUDIES

COMET was built from two studies of collective design. These two studies have used complementary methodological approaches. But, if different choices have been made at the outset, comparable results have been obtained. We give here some of the major results, summarized from (Falzon & Darses, 1992; Darses et al., 1993; Darses & Falzon, 1994; Darses et al., 1996; D'Astous et al., 1998; Robillard et al., 1998).

6.1 Description of the design case studies

Table 1 summarises the features of the two studies during which COMET has been built. Several facts deserve to be stressed. Both studies deal with real, complex design tasks. The two studies have been conducted independently and at different periods (several years between the two). However, the authors knew and read each other's publications.

Domain	*Computer network design*	*Software design*
Nature of the task	*Solution production*	*Solution review*
N° of designers	2	4
Domain of expertise	*Same expertise*: local area networks (LAN) configuration	*Same expertise*: software engineering
Level of expertise	*Different levels*: one experienced designer and one less competent designer	*Same levels*: all participants are competent designers
Situation	*Semi-experimental field study*: the situation of co-design was a usual practice. The less competent designer had to propose a solution to the network-configuration the problem that he was provided with. The instructions given to the expert demanded that he would collaborate with the less competent designer without taking charge of the problem. The two designers were not face to face but communicated by connected terminals. They shared a common representation of the network being designed, on which both could act.	*Field study*: the situation of co-design was technical-review meetings (TRMs) in an industrial software-development project. A TRM may occur after each phase in the global software-development process. It requires the presence of several reviewers. It has two main objectives: to verify the current state of the design project and to validate the specifications of the succeeding tasks. This is done on the basis of discussion of a document written in natural and/or programming language. Roles in these meetings are: manager of the meeting, manager of the project, "software-norms" expert/guarantor and presenter of the document (often author of the document under review).

6.2 Results obtained from applying COMET in network design

Results obtained from applying COMET in this case study were focused at understanding the expert's behaviour, regarding its assistance role. Would the expert provide the less experienced expert only with the "good" solution, or would he try to co-build the solution ?

Indeed, results highlight the large part of evaluative activity in collaborative design and the central role of criticism (positive, negative or mitigated) in the progression of cooperative reasoning. This evaluative activity is essentially due to the expert's contribution: the less experienced designer proposes an element of solution and the more experienced one reacts by evaluating the proposal. Positive evaluations may be followed by solution extensions (additions to the solution state) or by preventive statements (indications of not-to-be-forgotten facts or actions, implied by the solution proposal). Negative evaluations are always accompanied by justifications and often by alternative proposals —an observation also encountered in the reviews in our companion study. Moreover, the experienced designer also acts as a meta-planner, by proposing problems to be tackled or by suggesting to switch to a new design phase. The interventions of the experienced designer are not limited to solution proposals (as most classical expert systems behave): evaluation and planning are essential features of the experienced designer's behaviour. When he makes solution proposals, these are often part of a wider assistance move (e.g. negative evaluation).

The activity of evaluation has three characteristics:

- it is spontaneous: there are few explicit requests for evaluation on the part of the less experienced operator but much criticism formulated by the expert. This translates the action of conversionally specific rules to these situations of shared assistance in which the partners don't have the same level of knowledge in the speciality;
- it is backed: the evaluations are systematically accompanied by justifications. The role of these justifications is multiple: they specify the knowledge element on which the criticism is

grounded but they can play a role in the processes of cognitive synchronization. It must be noted that the justifications also accompany the positive evaluations in order to reinforce the choices made;

- it is extended: the evaluations are frequently prolonged by diverse extensions. In the case of negative evaluations, there are alternative propositions; in the case of "mitigated" evaluations, there are amendments or preventive information ("that can work, but then we will have to think about ..."); in the case of positive evaluations, there are additions of new elements in the proposed solution.

Results indicate also that the design of a given stage of a solution - to generate the solution, evaluate the solution, to plan, etc. - , is not always taken over by the same operator, even when neither one of them has the same level of ability.

6.3 Results obtained from applying COMET in software reviewing

In technical-review meetings (TRMs), we found that, when introduction of a solution is followed immediately by a development, this development consists in changing the form of the solution, and there is an implicit negative evaluation according to a criterion of form. Introduction of a solution can also be followed immediately by either its evaluation alone or its evaluation and development of an alternative solution (in one order or another). Such review activities may, or may not, be preceded by a cognitive-synchronisation exchange. If they are, the evaluation bears mainly upon content criteria.

With respect to the relationship between review and cognitive synchronisation: when review is introduced by cognitive synchronisation, this means that a shared representation of the to-be-evaluated object may be a prerequisite for its review to take place. The argumentative movement is of the type “proposition-opinion”.

With respect to the relationship between review and design: the review of a solution, in particular a negative review, often leads participants to make explicit alternative solutions: such a solution may be a justification for the negative review or a solution for the current rejection. The argumentative movement is of the type “opinion-arguments” (On the surface, the arguments may be presented, either before the opinion it supports is presented, or afterwards).

We thus notice that the activities of elaboration and of cognitive synchronisation, even if not expected in the prescribed task, are both necessary and useful in the collaboration that takes place through argumentation in TRMs.

7 CONCLUSION

7.1 Improving the method COMET

Our goal is to gradually build a consistent method of analysis of cooperative design dialogues. The fact that COMET has been successfully used in two different design situations is promising. COMET has already been (or is currently) tested in other situations (aeronautical design, organizational negotiation, participatory design in hospitals, etc.). These tests will be used to improve the generic nature of the method. During this process, particular attention should be paid to the production of precise definitions of predicates and arguments and of rules of specifications of the CO-OPERATION MOVES.

The benefits of COMET are:

- a quicker and systematic encoding of verbal data. For the time being, researchers in the design field produce *ad hoc* method of encoding, for each corpus they consider. This situation is time-consuming and prone to errors, and does not allow analysts to compare their results.
- a flexible exploitation of the basic coding scheme. The basic level of encoding in elementary UNITS can be used in various ways, since it avoids the encoding of argumentation.

COMET would be easier to use if it were supported by a software allowing to segment the utterances, to encode them, and to test various groupings of UNITS into sequences. Such a system could be developed specifically, or on the basis of existing software like MacSHAPA (Sanderson et al., 1994) or KRONOS (Kerguelen, 1989 & 1998).

7.2 COMET as a CTA method for design processes

A first result obtained through the application of COMET concerned the nature of the contributions to design problem solving. Solution generation is often thought of as the main contribution to problem solving in design, probably because the final state itself is described as a solution. The analysis of elementary UNITS shows that many contributions are necessary in order to reach a satisfying solution. Focus generation, evaluation, problem description are contributions as important as solution generation.

A second result concerns the mechanisms of collaborative design. The analysis of sequences (which go towards the identification of the CO-OPERATION MOVES) makes it possible to describe the way in which collective processing proceeds. This description shows how all contributions to problem solving are based on previous interventions of the partners, which is consistent with an interactional model of dialogue processes.

The ultimate goal of studies of design dialogues is the description of the conditions of collective performance. COMET allows the analyst to show what is actually produced by the group (e.g. solutions, but also justifications, criteria, pending problems, etc.). A difficulty in stating the conditions of performance lies in the fact that performance itself is difficult to assess: in general, comparisons are impossible, at least in real-life situations. Experimental work may be useful in that perspective.

COMET was developed for analysing co-design processes occurring in the same place, where the designers are present in the same room. It can likely also be applied for design work mediated by communication tools (such as web-based tools), where the designers do not design at the same time. The growing importance of such cooperative design situations makes COMET a good candidate a CTA methods usable for such cooperative design environments.

8 REFERENCES

Allison, P.D., Liker, J.K. (1982). Analysing Sequential Categorical Data on Dyadic Interaction: A comment on Gottman. *Psychology Bulletin, 91*, 393-403.

Amalberti, R., & Hoc, J. M. (1998) Cognitive activity analysis in dynamic situation: why and how ? *Le Travail Humain, 61 (3)*, 209-234.

Austin, J. L. (1962) *How to Do Things with Words.* J.O. Urmson (Ed.) Oxford: Clarendon Press.

Béguin, P. (1994). Travailler avec la CAO en ingénierie industrielle: de l'individuel au collectif dans les activités avec instruments. *Thèse de Doctorat d'Ergonomie*, C.N.A.M., Paris.

Brassac, C. & Grégori, N. (2001) Eléments pour une clinique de la conception collaborative. *Proceedings of 10th Workshop of Le Travail Humain*, Paris, 27-28 June, Ed. INRIA.

Brassac, C. (1992) Analyse de conversations et théorie des actes de langage. *Cahiers de linguistique française, 13*, 62-76

Brinkman, J. A. (1994) Methological problems when determining verbal protocol accuracy empirically. *Le Travail Humain, 57 (2)*, 97-109.

Cahour, B. (2001) Décalages socio-cognitifs en réunions de conception industrielle *(Socio-cognitive gaps in design meetings). Proceedings of the 10th Workshop of Le Travail Humain, Paris, 27-28 June, INRIA.*

Cahour, B. & Falzon, P. (1991). Assistance à l'opérateur et modélisation de sa compétence. *Intellectica*, 2, 12, 159-186.

Cross, N. Christiaans, H. & Dorst, K., (eds) (1996) *Analysing design activity* (N.Y.: J.Wiley and Sons).

D'Astous, P., Détienne, F., Robillard, P. N., & Visser, W. (1998). Types of dialogues in evaluation meetings: an analysis of technical-review meetings in software development. *Proceedings of COOP'98, Third International Conference on the Design of Co-operative Systems*, Cannes (France), 26-29 May, 1998.

Darses, F. & Falzon, P. (1994). Activités de coopération au cours de la résolution de problèmes de conception. *Workshop sur la Coopération à la Conférence Internationale sur les Systèmes Experts et leurs Applications*, Paris, 30-31 May 1994.

Darses, F. & Falzon, P. (1996). La conception collective: une approche de l'ergonomie cognitive. In G. de Terssac et E. Friedberg (Eds.). *Coopération et Conception.* Octarès: Toulouse.

Darses, F., Falzon, P. & Béguin, P. (Eds.) (1996). Collective design processes. *Proceedings of COOP 96, Second International Conference on the Design of Co-operative Systems*, Juan-les-Pins (France), 12-14 juin, Sophia-Antipolis: INRIA.

Darses, F., Falzon, P. & Robert, J.M. (1993). Cooperating partners: investigating natural assistance. In G. Salvendy & M.J. Smith (Eds.), *Human-computer interaction: Software and hardware interfaces*. New York: Elsevier.

Détienne, F., Visser, W., D'Astous, P., & Robillard, P. N. (1999). Two complementary approaches in the analysis of design team work: the functional and the interactional approach. *Position Paper presented at CHI99 Basic Research Symposium*, Pittsburgh, Pennsylvania, USA, May 15-16 1999.

Dorst, K. (1995) (Ed). Analysing design activity. *Special Issue of Design Studies, 16 (2).*

Ericsson, K. A. & Simon, H. (1984). *Protocol analysis. Verbal reports as data.* USA: MIT Press.

Falzon, P. & Darses, F. (1992). Les processus de coopération dans les dialogues d'assistance. *Actes du XXVII° Congrès de la SELF*. Lille, 23-25 septembre 1992.

Falzon, P. (1991). Co-operative dialogues. In J. Rasmussen, J. Leplat & B. Brehmer (Eds.) *Distributed decision-making*. Chichester: Wiley.

Ghiglione, R. & Trognon, A. (1993) *Où va la pragmatique ? De la pragmatique à la psychologie sociale.* Grenoble: Presses Universitaires de Grenoble.

Goel, V. & Pirolli, P. (1989). Motivating the notion of generic design within information-processing theory: the design problem space. *AI Magazine, Spring*, 18-36.

Goldsmidt, G. (1995) The designer as a team of one. In K. Dorst (ed) *Analysing design activity. Special Issue of Design Studies, 16 (2*

Gonzalez, R.C., Thomason, M.G. (1978). *Syntactic Pattern Recognition: An Introduction.* Addison-Wesley.

Hoc, J. M. & Amalberti, R. (1999) Cognitive activity analysis in dynamic situation: from theoretical framework to method? *Le Travail Humain, 62 (2)*, 97-129.

Hoc, J. M. (2000) From human-machine interaction to human-machine cooperation. *Ergonomics, 43*, 833-843.

Johnson, S.C. (1967). Hierarchical Clustering Schemes. *Psychometrika, 32*, 241-254.

Karsenty, L. (1994). L'explication d'une solution dans des dialogues de conception. *Thèse de Doctorat de Psychologie Cognitive Ergonomique*, Université Paris 8.

Karsenty, L. & Falzon, P. (1992). Cadre général pour l'étude des dialogues orientés-tâche. *Actes du XXVII° du Congrès de la SELF*. Lille, 23-25 septembre.

Kerguelen, A. (1989) KRONOS pour Psion Organiser. Lyon: ANACT.

Lebahar, J.C. (1983). *Le dessin d'architecte*. Roquevaire: Parenthèses.

Malhotra, A., Thomas, J. Carroll, J., & Miller, L. (1980). Cognitive processes in design. *International Journal of Man-Machine Studies*, 12 (119-140).

Martin, G., Détienne, F., & Lavigne, E. (2001, July 9-13). *Analysing viewpoints in design through the argumentation process*. Proceedings of Interact 2001, Tokyo, Japan.

Newell, A. & Simon, H. (1972). *Human Problem Solving*. Englewood Cliffs, N. J.: Lawrence Erlbaum.

Reuzeau, F. (2000) Assiser l'évaluation participative des systèmes complexes: rôle des savoirs et savoir-faire des utilisateurs dans la conception d'un poste de pilotage d'avions. Thèse de Doctorat, Spécialité Ergonomie, CNAM, Paris.

Robillard, P. N., D'Astous, P., Détienne, F., & Visser, W. (1998). Measuring cognitive activities in software engineering. *Proc. of ICSE98, 20th International Conference on Software Engineering*, Kyoto (Japan), April 19-25, 1998.

Sanderson, P. , Scott, J., Johnston, T. , Mainzer, J. , Watanabe, L. & James, J. (1994) MacSHAPA and the enterprise of exploratory sequential data analysis. *International Journal of Human-Computer Studies, 41 (5),* 633-681.

Sauvagnac, C. & Falzon, P. (1996) Collaboration and underlying issues or the surprises of cooperative dialogues, CSCW, 5, 251-266.

Searle, J. R. (1969) *Speech acts*. London: Cambridge Press University.

Simon, H. (1981). The structure of ill-defined problems. *Artificial Intelligence*, *4*, 181-201.

Sperber, D & Wilson,1986) Relevance. Communication and cognition. Oxford: Blackwell.

Terssac (de), G. & Chabaud, C. (1990). Référentiel opératif commun et fiabilité. In J. Leplat et G. de Terssac (Eds.) *Les facteurs humains de la fiabilité dans les systèmes complexes*. Paris: Octarès.

Visser, W. (1991). Evocation and elaboration of solutions: Different types of problem-solving actions. An empirical study on the design of an aerospace artifact. In T. Kohonen and F. Fogelman-Soulié (Eds.), *COGNITIVA 90. At the crossroads of Artificial Intelligence, Cognitive science, and Neuroscience. Proceedings of the third COGNITIVA symposium* . Amsterdam: Elsevier.

Visser, W. (1992). Designers' activities examined at three levels: organization, strategies & problem-solving. *Knowledge-Based Systems*, *5*, 1, 92-104.

Visser, W. (1993). Collective design: A cognitive analysis of co-operation in practice. In N. F. M. Roozenburg (Ed.). *Proceedings of ICED 93. 9th International Conference on Engineering Design* (Volume 1), The Hague, August 17-19, 1993. Zürich: HEURISTA

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399

ISRN INRIA/RR--4258--FR+ENG

RR - 4258